STUDI SEJARAH DAN PERKEMBANGAN BILANGAN PRIMA

Jansen - NIM : 13506028
Jurusan Teknik Informatika ITB, Bandung
email: if16028@students.if.itb.ac.id

Abstract *– Makalah ini membahas tentang perkembangan salah satu topik dalam teori bilangan, bilangan prima dan perkembangan algoritma-algoritma dalam masalah yang berkaitan bilangan prima. Bilangan prima adalah sebuah bilangan lebih besar dari satu yang hanya terdiri dari dua faktor, yaitu bilangan 1 (satu) dan bilangan itu sendiri. Keadaan sebuah bilangan adalah prima disebut primalitas. Contoh penggunaan bilangan prima adalah pada kriptografi sebagai kunci umum (public key).*
Studi tentang bilangan prima telah menjadi bagian dari salah satu cabang matematika, yaitu Number Theory. Masalah bilangan prima telah menjadi salah satu bahasan pokok dari riset-riset yang dilakukan di dunia saat ini.
Dari riset-riset yang telah dilakukan sebelumnya, ditemukan beberapa teknik dalam pencarian apakah sebuah bilangan adalah bilangan prima atau bukan, seperti Fermat Little Test.
Namun dalam hal pencarian pola deret bilangan prima tidaklah semudah itu. Definisi bilangan prima yang sederhana tersebut tidak menjamin bahwa bilangan prima muncul dengan pola-pola yang teratur, bahkan sebaliknya; tidak ada yang dapat memprediksi kemunculan bilangan prima yang selanjutnya seperti yang telah ditunjukkan oleh Euclid sekitar tahun 300 SM.
Untuk mencari deret bilangan prima, dapat dilakukan dengan salah satunya sieve of Eratosthenes.

Kata Kunci*:* bilangan prima, *Mersenne prime, Twin prime, Fermat's prime, Sieve of Eratosthenes, Sieve of Atkin,* persamaan Diophantine, kompleksitas, algoritma.

1. PENDAHULUAN

Bilangan prima adalah bilangan bulat positif lebih besar dari satu yang memiliki tepat hanya dua faktor, yaitu bilangan 1 (satu) dan bilangan itu sendiri. Bilangan selain bilangan prima - yang memiliki lebih dari dua faktor - dan selain bilangan 1 (satu) disebut sebagai bilangan komposit. Bilangan 1 adalah kasus khusus, tidak termasuk ke bilangan prima ataupun komposit. Banyak orang menanyakan sebabnya. Walau bilangan 1 dulunya dianggap sebagai bilangan prima (Goldbach 1742; Lehmer 1909; Lehmer 1914; Hardy and Wright 1979, hal. 11; Gardner 1984, hal. 86-87; Sloane and Plouffe 1995, hal. 33; Hardy 1999, hal. 46), diperlukan penanganan khusus di banyak definisi dan aplikasi yang mencakup bilangan prima, sehingga bilangan 1 tersebut biasanya diletakkan pada sebuah kelas khusus.
Pengunaan bilangan prima lebih lanjut dapat ditemukan pada penggunaan *public key* pada metode kriptografi.
Primalitas adalah keadaan di mana sebuah bilangan termasuk bilangan prima. Bilangan prima telah menjadi subyek dari riset-riset di dunia dan bahkan telah menjadi pertanyaan mendasar, seperti Hipotesa Riemann dan Dugaan Goldbach, yang tetap tidak terpecahkan selama lebih dari satu abad.

2. VARIASI JENIS BILANGAN PRIMA

2.1. *Mersenne prime*

Mersenne prime adalah bilangan Mersenne yang juga adalah bilangan prima. Bilangan Mersenne memiliki rumus :

$$M_n = 2^n - 1 \qquad (1)$$

n	Mn	Jumlah digit Mn	Tanggal ditemukan	Penemu
2	3	1	??	??
3	7	1	??	??
5	31	2	??	??
7	127	3	??	??
13	8191	4	1456	??
17	131071	6	1588	Cataldi
19	524287	6	1588	Cataldi

Tabel berikut adalah daftar dari beberapa bilangan prima Mersenne pertama yang sudah ditemukan.

2.2. *Fermat prime*

Fermat prime adalah bilangan Fermat yang juga adalah bilangan prima. Bilangan Fermat memiliki rumus :

$$F_n = 2^{2^n} + 1 \qquad (2)$$

2.3. *Twin prime*

Twin prime (prima kembar) adalah bilangan prima yang berasal dari penggabungan/konkatenansi dua buah bilangan prima yang berselisihkan tepat 2. Beberapa contoh pasangan bilangan prima kembar adalah (3, 5), (5, 7), (11, 13), (17, 19), (29, 31),

(41, 43), (59, 61), (71, 73), (101, 103), (107, 109), (137, 139), (149, 151), (179, 181), (191, 193), (197, 199), (227, 229), dan (239, 241).

3. BILANGAN PRIMA TERBESAR

Bilangan prima Mersenne yang terbesar saat ini adalah $2^{32582657}$-1 yang memiliki 9.808.358 digit, ditemukan tahun 2006 oleh Dr. Curtis Cooper and Dr. Steven Boone dan sekaligus merupakan bilangan prima Mersenne ke-44.

Bilangan prima Fermat terbesar saat ini adalah 65.537, yang merupakan bilangan Fermat ke-4. Perlu diketahui sampai saat bahwa hanya F_0, F_1, F_2, F_3 dan F_4 yang merupakan bilangan prima Fermat.

Pasangan bilangan prima kembar terbesar saat ini adalah $2003663613 \cdot 2^{195000} \pm 1$ yang masing-masing memiliki 58711 digit, ditemukan pada 15 Januari 2007 oleh Eric Vautier dari Prancis.

Bilangan prima yang terbesar saat ini adalah bilangan Mersenne ke-44 tersebut. Sepanjang sejarah penemuan, bilangan prima terbesar yang ditemukan sebagian besar selalu merupakan bilangan prima Mersenne terbesar, dengan beberapa pengecualian seperti pada periode antara penemuan Brown bulan Agustus 1989 dan penemuan Slowinski dan Gage bulan Maret 1992.

4. PENGETESAN PRIMALITAS

Pengetesan primalitas sebuah bilangan dapat menggunakan beberapa teknik (selain cara awam), contohnya adalah menggunakan *Fermat Little Test*.

4.1. Metode awam

Dalam perkembangannya, cara awam mengalami beberapa perbaikan dalam hal efisiensi dan keefektifannya.

4.1.1. Metode awam (sebelum perbaikan)

Cara awam adalah cara pengetesan primalitas sebuah bilangan menurut definisi bilangan prima itu sendiri.

4.1.1.1. Algoritma

Diberikan sebuah bilangan, *n*, jika *n* dapat membagi salah satu bilangan dari 2 sampai *n*-1, maka *n* dikatakan bukan bilangan prima, dan sebaliknya.

```
1   isPrime(n)
2     checkPrime ← true
3     i ← 2
4     while (i = n-1) and (checkPrime) do
5       if (n mod i = 0) then
6         checkPrime = false
7       i ← i + 1
7     return (checkPrime)
```

Pseudocode Cara Awam (sebelum perbaikan)

4.1.1.2. Analisis

Metode awam sebelum perbaikan ini menggunakan sebuah iterasi `while` untuk mengecek segala kemungkinan terbaginya *n* dari 2 sampai (*n*-1).

```
while (i = n-1) and (checkPrime) do    O(n)
    if (n mod i = 0) then              O(1)
       checkPrime = false              O(1)
    i ← i + 1                          O(1)
```

$$T(n) = O(n)(O(1) + O(1) + O(1))$$
$$T(n) = O(n)O(max(1,1,1)) = O(n)O(1)$$
$$T(n) = O(n .1)$$
$$T(n) = O(n)$$

Total waktu yang digunakan adalah **O(n)**.

4.1.2. Metode awam (perbaikan 1)

Dikarenakan tidak efektifnya pencarian dapat terbagi tidaknya bilangan tersebut oleh banyaknya bilangan, maka perbaikan yang dilakukan adalah dengan membatasi jumlah pencarian.

4.1.2.1. Algoritma

Pembatasan jumlah pencarian adalah dari 2 sampai $\sqrt{n}$ karena jika n adalah bilangan komposit, maka n dapat difaktorkan menjadi dua nilai, akan tetapi salah satu nilai tersebut pastilah lebih kecil atau sama dengan $\sqrt{n}$.

```
1   isPrime(n)
2     checkPrime ← true
3     i ← 2
4     while (i = sqrt(n)) and (checkPrime) do
5       if (n mod i = 0) then
6         checkPrime = false
7       i ← i + 1
8     return (checkPrime)
```

Pseudocode Cara Awam (perbaikan 1)

4.1.2.2. Analisis

Metode awam setelah perbaikan pertama ini menggunakan sebuah `while` untuk mengecek segala kemungkinan terbaginya *n* dari 2 sampai $\sqrt{n}$.

```
while (i = sqrt(n)) and (checkPrime) do    O(log n)
    if (n mod i = 0) then                  O(1)
       checkPrime = false                  O(1)
    i ← i + 1                              O(1)
```

$$T(n) = O(log\ n)(O(1) + O(1) + O(1))$$
$$T(n) = O(log\ n)O(max(1,1,1)) = O(log\ n)O(1)$$
$$T(n) = O(log\ n .1)$$
$$T(n) = O(log\ n)$$

Total waktu yang digunakan adalah **O(log n)**.

4.1.3. Metode awam (perbaikan 2)

Dengan mempertimbangkan fakta bahwa bilangan genap yang merupakan bilangan prima hanyalah bilangan 2, maka pencarian terbaginya bilangan *n* dapat disederhanakan.

4.1.3.1. Algoritma

Jika bilangan tersebut adalah bilangan 2, maka pengetesan bilangan prima adalah benar.

Jika bilangan tersebut bilangan genap selain bilangan 2, maka pengetesan bilangan prima adalah salah.

Jika selain bilangan (bilangan ganjil), pencarian dilakukan dengan metode awam perbaikan 1, namun iterasi yang dilakukan adalah untuk bilangan ganjil.

```
1   isPrime(n)
2      if (i = 2) then
3         checkPrime ← true
4      else if (i mod 2 = 0) then
5         checkPrime ← false
6      else
7         checkPrime ← true
8         i ← 3
9         while (i = sqrt(n)) and (checkPrime) do
10           if (n mod i = 0) then
11              checkPrime = false
12           i ← i + 2
13        return (checkPrime)
```

Pseudocode Cara Awam (Perbaikan 2)

4.1.3.2. Analisis

Metode awam setelah perbaikan kedua ini tetap menggunakan sebuah `while`.

```
while (i = sqrt(n)) and (checkPrime) do    O(log n)
   if (n mod i = 0) then                   O(1)
      checkPrime = false                   O(1)
   i ← i + 2                               O(1)
```

$$T(n) = O(\log n)(O(1) + O(1) + O(1))$$
$$T(n) = O(\log n)O(\max(1,1,1)) = O(\log n)O(1)$$
$$T(n) = O(\log n .1)$$
$$T(n) = O(\log n)$$

Total waktu yang dipakai adalah tetap **O(log n)**, namun dengan ditambahkannya `if` untuk prekomputasi yang membuat kemungkinan algoritma ini menghasilkan kasus terbaik, **O(1)**, lebih besar**.**

4.2. *Fermat Little Test*

Ini adalah algoritma yang "rancu", di mana tidak dapat dipastikan kemungkinan hasil pengetesan bilangan prima adalah benar. Dalam jangkauan bilangan 1 – 1.000.000, Fermat Little Test dapat mengacu kepada hasil yang salah untuk 255 bilangan Carmichael (tidak akan dibahas di makalah ini). Akan tetapi, dapat dilakukan beberapa pengecekan asal untuk meningkatkan kemungkinan tersebut.

Fermat Little Test adalah sebagai berikut:
Jika 2^n modulo n = 2, maka n mempunyai kemungkinan besar untuk menjadi bilangan prima.

Sebagai contoh,
N = 3,
2^3 mod 3 = 8 mod 3 = 2, maka N mempunyai kemungkinan besar untuk menjadi bilangan prima, bahkan memang N adalah bilangan prima.

Contoh lainnya,
N = 11,
2^{11} mod 11 = 2048 mod 11 = 2, maka N mempunyai kemungkinan besar untuk menjadi bilangan prima, dan memang N adalah bilangan prima.

5. DERET BILANGAN PRIMA

Dalam sejarah perkembangan bilangan prima, telah ditemukan beberapa algoritma dalam pencarian deret bilangan prima, seperti *Sieve of Eratosthenes* dan *Sieve of Atkin*.

5.1. *Sieve of Eratosthenes*

Algoritma yang paling sederhana dan paling tua mungkin adalah *Sieve of Eratosthenes*. Algoritma ini ditemukan oleh matematikawan Yunani kuno, Eratosthenes.
Algoritma *Sieve of Eratosthenes* adalah sebagai berikut:

1. Pertama-tama, tuliskan daftar bilangan dari 2 sampai batas atas bilangan yang akan dicari.
2. Kemudian, tandai bilangan di dalam daftar yang merupakan kelipatan 2, dengan meninggalkan bilangan 2 tetap tidak ditandai.
3. Lanjutkan ke bilangan berikutnya (dalam tahap ini adalah bilangan 3), dan tandai setiap kelipatan 3, dengan tetap meninggalkan bilangan 3 tidak ditandai.
4. Lanjutkan ke bilangan berikutnya. Bila bilangan berikut tersebut telah ditandai, lanjutkan ke bilangan berikutnya.
5. Lanjutkan langkah penandaan seperti di atas sampai batas atas bilangan.

Bilangan yang belum ditandai adalah merupakan bilangan prima.
Algoritma ini menggunakan fakta bahwa jika sebuah bilangan adalah prima, maka kelipatan dari bilangan tersebut pastilah bukan prima.
Algoritma ini mempunyai kompleksitas algoritma **O(N^2)** dengan *N* adalah batas atas bilangan.

Ilustrasi:
Misal batas atas = 15,
daftar bilangan yang dibuat adalah

2	3	4	5	6	7	8	9	10	11	12	13	14	15

Tandai kelipatan setiap bilangan yang merupakan kelipatan 2 (bilangan 2 tidak ditandai).

2	3	4	5	6	7	8	9	10	11	12	13	14	15

Lanjutkan ke bilangan 3 dan tandai setiap kelipatan 3, bilangan 3 tidak ditandai.

2	3	4	5	6	7	8	9	10	11	12	13	14	15

Karena bilangan 4 sudah ditandai, lanjut ke bilangan 5, dan tetap ditandai kelipatannya.

2	3	4	5	6	7	8	9	10	11	12	13	14	15

Lanjut ke bilangan 7, seterusnya sampai bilangan 15.

2	3	4	5	6	7	8	9	10	11	12	13	14	15

Dari tabel tersebut, didapatkan bahwa bilangan prima

dengan batas atas 15 adalah {2, 3, 5, 7, 11, 13}.

5.2. *Sieve of Atkin*

Sieve of Atkin merupakan algoritma pengembangan dari *Sieve of Eratosthenes*. Algoritma ini ditemukan oleh A. O. L. Atkin dan Daniel J. Bernstein.

Dalam algoritma ini:

- semua sisa bilangan adalah hasil modulo bilangan tersebut dengan 60,
- semua bilangan, termasuk x dan y, adalah bilangan bulat positif, serta
- membalikkan sebuah entri dalam daftar "penyaringan" berarti membalikkan penandaan tentang primalitas bilangan tersebut ke keadaan sebaliknya (prima ke bukan prima, dan sebaliknya).

Algoritma *Sieve of Atkin* adalah sebagai berikut:

1. Buatlah sebuah daftar hasil, isi dengan 2, 3, dan 5.
2. Buatlah daftar "penyaringan" dengan sebuah entri untuk setiap bilangan bulat positif; semua entri pada daftar ini telah sebelumnya ditandai bukan bilangan prima.
3. Untuk setiap entri di daftar "penyaringan" :
   - Jika entri tersebut adalah sebuah bilangan dengan sisa 1, 13,17, 29, 37, 41, 49, atau 53, balikkan entri bilangan tersebut untuk setiap kemungkinan solusi $4x^2 + y^2$ = bilangan_entri.
   - Jika entri tersebut adalah bilangan dengan sisa 7, 19, 31, atau 43, balikkan entri bilangan tersebut untuk setiap kemungkinan solusi $3x^2 + y^2$ = bilangan_entri.
   - Jika entri tersebut adalah bilangan dengan sisa 11, 23, 47, atau 59, balikkan entri bilangan tersebut untuk setiap kemungkinan solusi $3x^2 - y^2$ = bilangan_entri untuk $x > y$.
   - Abaikan entri yang memiliki sisa selain di atas.
4. Mulai dengan bilangan paling kecil pada daftar "penyaringan".
5. Ambil bilangan berikutnya yang terdapat pada daftar "penyaringan" yang masih ditandai prima.
6. Masukkan bilangan tersebut pada daftar hasil.
7. Kuadratkan bilangan tersebut dan tandai semua kelipatan dari hasil kuadrat tersebut sebagai bilangan bukan prima.
8. Ulangi langkah lima sampai langkah delapan.

Algoritma ini mengkomputasikan bilangan-bilangan prima sampai N dengan kompleksitas $\mathbf{O(}N\mathbf{/log\ log}\ N\mathbf{)}$ operasi dan hanya menggunakan $N^{1/2+O(1)}$ jumlah memori.

6. RUMUSAN DERET BILANGAN PRIMA

Belum ditemukan adanya rumusan deret bilangan prima yang tepat dan lebih efisien dibandingkan menggunakan metode-metode pengecekan primalitas bilangan di atas untuk tiap bilangan secara iteratif dalam hal pencarian bilangan prima. Selain itu, ada tidaknya rumusan deret bilangan prima yang tepat masih diragukan.

Telah didapatkan bahwa tidak ada fungsi polinomial $P(n)$ yang non-konstan yang mengecek sampai pada sebuah bilangan prima untuk setiap bilangan n. Sebagai bukti: Asumsikan fungsi polinomial tersebut ada. Lalu $P(1)$ akan mengecek sampai pada sebuah bilangan prima p, sehingga $P(1) = 0 \pmod p$. Akan tetapi untuk semua nilai k, $P(1 + kp) = 0 \pmod p$ juga, sehingga $P(1 + kp)$ bukanlah sebuah bilangan prima (karena dapat dibagi oleh p) kecuali adalah p sendiri, tetapi satu-satunya cara $P(1 + kp) = P(1)$ untuk semua nilai k adalah jika fungsi polinomial tersebut konstan.

Dengan menggunakan teori-teori bilangan yang lebih "aljabar", akan terlihat lebih jelas akan tidak adanya fungsi polinomial $P(n)$ yang bersifat non-konstan yang mengecek sampai pada sebuah bilangan prima untuk hampit setiap bilangan n.

Walau rumusan deret bilangan prima belum ditemukan, namun ada sebuah kumpulan persamaan yang dinamakan persamaan Diophantine dalam 26 buah peubah-peubah yang dapat digunakan untuk mendapatkan bilangan-bilangan prima walau tidak efisien. Jones (1976) telah membuktikan bahwa bilangan $k + 2$ adalah prima jika dan hanya jika ke-14 persamaan Diophantine berikut mempunyai hasil solusi dalam bilangan natural:

$0 = wz + h + j - q$
$0 = (gk + 2g + k + 1)(h + j) + h - z$
$0 = 16(k + 1)^3(k + 2)(n + 1)^2 + 1 - f^2$
$0 = 2n + p + q + z - e$
$0 = e^3(e + 2)(a + 1)^2 + 1 - o^2$
$0 = (a^2 - 1)y^2 + 1 - x^2$
$0 = 16r^2y^4(a^2 - 1) + 1 - u^2$
$0 = n + l + v - y$
$0 = (a^2 - 1)l^2 + 1 - m^2$
$0 = ai + k + 1 - l - i$
$0 = ((a + u^2(u^2 - a))^2 - 1)(n + 4dy)^2 + 1 - (x + cu)^2$
$0 = p + l(a - n - 1) + b(2an + 2a - n^2 - 2n - 2) - m$
$0 = q + y(a - p - 1) + s(2ap + 2a - p^2 - 2p - 2) - x$
$0 = z + pl(a - p) + t(2ap - p^2 - 1) - pm.$

Persamaan ini dapat digunakan untuk menghasilkan polinom yang selanjutnya akan menghasilkan bilangan-bilangan prima. Misalkan sisi kanan dari persamaan-persamaan tersebut di atas adalah a_1, …, a_{14}. Maka persamaan

$$(k+2)(1-\alpha_1^2-\alpha_2^2-\cdots-\alpha_{14}^2)$$

adalah sebuah polinom dalam 26 variabel, dan kumpulan bilangan-bilangan prima adalah identik dengan kumpulan nilai-nilai positif yang dihasilkan polinom di atas sementara peubah-peubah a, b, …, z juga memiliki jangkauan nilai bilangan bulat bukan negatif.

7. HADIAH PENEMUAN BILANGAN PRIMA

Electronic Frontier Foundation (EFF) telah menawarkan hadiah US$100,000 untuk penemu pertama bilangan prima dengan jumlah digit minimum 10 juta buah. Mereka juga menawarkan hadiah US$150,000 untuk 100 juta buah jumlah digit bilangan prima, dan US$250,000 untuk 1 milyar buah jumlah digit bilangan prima. Pada tahun 2000 kemaren, EFF memberikan hadiah US$50,000 untuk 1

juta jumlah digit bilangan prima yang ditemukan.

8. KESIMPULAN

Teori bilangan adalah studi-studi tentang bilangan. Salah satu contohnya adalah bilangan prima.
Bilangan prima, secara terpisah, memiliki berbagai jenis penurunan, seperti bilangan prima Mersenne, bilangan prima Fermat dan bilangan prima kembar.
Pengecekan bilangan prima dapat dilakukan dengan cara sederhana ataupun menggunakan *Fermat Little Test*. Perlu diketahui bahwa belum ada formula yang secara tepat mengecek primalitas suatu bilangan.
Untuk mencari deret bilangan prima, dapat digunakan *Sieve of Eratosthenes* dan *Sieve of Atkin* yang merupakan pengembangan dari *Sieve of Eratosthenes*.
Belum ditemukan adanya rumusan yang mangkus dalam mencari deret bilangan prima.

DAFTAR REFERENSI

[1] Wikipedia. http://en.wikipedia.org/wiki/Prime_number; Tanggal akses pada 27 Desember 2007.

[2] Mathworld, Wolfram. http://mathworld.wolfram.com/topics/PrimeNumbers; Tanggal akses pada 27 Desember 2007.

[3] sci.math, FAQS. http://www.faqs.org/faqs/sci-math-faq/primes; Tanggal akses pada 27 Desember 2007.

[4] Mersenne. http://www.mersenne.org/prime.htm; Tanggal akses 28 Desember 2007.

[5] Steven S. Skiena, Miguel A. Revilla. *Programming Challenges*. Springer-Verlag New York, Inc., 2003. ISBN 0-387-00163-8.

[6] Richard Crandall, Carl Pomerance. *Prime Numbers: A Computational Perspective*. Springer Science+Business Media, Inc., 2005. ISBN 0-387-25282-7.

[7] David Wells. *Prime Numbers: The Most Mysterious Figures in Math*. John Wiley & Sons, Inc., 2005. ISBN 0-471-46234-9.